Claims

We claim:

1   1. A method for maintaining a system for database management, the method comprising:

2   during splitting of a leaf block of a database index recording an address of a newly

3   created leaf block; and

4   maintaining the new address in a list as part of metadata of a primary B+tree.

1   2. The method according to claim 1, further comprising:

2   maintaining a measure of invalid guess-database block addresses by calculating a ratio of

3   a count of database block addresses in the list of new addresses to a total number of leaf blocks

4   of the primary B+tree.

1   3. The method according to claim 2, wherein the measure of invalid guess-database

2   block addresses applies to mapping tables and secondary indexes on the primary B+tree.

1   4. The method according to claim 2, wherein the list of database block addresses and the

2   ratio are maintained only when the ratio is less than a threshold value.

1   5. The method according to claim 4, wherein the threshold value for the ratio is about

2   10%.

1   6. The method according to claim 3, further comprising:

2   adjusting a guess-DBA quality of at least one of the mapping table and the secondary

3 index utilizing the ratio.

1   7. The method according to claim 4, wherein if the ratio is below the threshold value the

2 method further comprises:

3   selectively correcting entries in the mapping table and/or secondary index.

1   8. The method according to claim 7, wherein correcting entries in the mapping table

2 comprises for all rows in a list of blocks in the primary B+tree:

3   obtaining corresponding mapping table row identifiers and database block addresses of a

4 current block in the list;

5   sorting the corresponding mapping table row identifiers;

6   obtaining mapping table rows corresponding to the mapping table row identifiers; and

7 updating a guess-DBA component if it has changed.

1   9. The method according to claim 8, wherein the correcting is carried out on-line in a

2 piece-wise manner.

1   10. The method according to claim 4, wherein correcting entries in the secondary index

2 comprises for all rows in a list of blocks in the primary B+tree:

3   obtaining a secondary index key, a primary key and a database block address of a current

4 block in the list of blocks;

13

5 sorting the secondary index keys, primary keys and database addresses in order of

6 (secondary index key, primary key) pairs;

7 obtaining an index row corresponding to the (secondary index key, primary key) pair; and

8 updating a guess-DBA component of the index row if the guess-DBA has changed.

1 11. The method according to claim 11, wherein the correcting is carried out on-line in a

2 piece-wise manner.

1 12. The method according to claim 4, wherein if the ratio is above the threshold value

2 the method further comprises:

3 correcting guess-database addresses on a per object basis.

1 13. The method according to claim 12, wherein correcting guess-database block

2 addresses on the mapping table comprises:

3 performing a full scan of the mapping table;

4 determining for each row of the mapping table a correct guess-database block address by

5 traversing the primary B+tree up to a penultimate level;

6 updating each row of the mapping table with the correct guess-database block address;

7 and

8 committing the correct guess-database address to the mapping table in batches.

1 14. The method according to claim 12, wherein correcting guess-database block

2 addresses on a per object basis comprises for each secondary index object:

14

3   performing a full scan of the secondary index object;

4   determining for each row of the secondary index a correct guess-database block address

5 by traversing the primary B+tree up to a penultimate level;

6   updating each row of the secondary index with the correct guess-database block address;

7 and

8   committing the correct guess-database block address to the secondary index in batches.

1   15. The method according to claim 1, further comprising:

2   maintaining a list of database block addresses in the list.

1   16. A system for organizing a database index, the system comprising:

2   a list of addresses of blocks newly created during splitting of a primary B+tree.

1   17. The system according to claim 16, further comprising:

2   a count of database block addresses in the list.

1   18. The system according to claim 16, further comprising:

2   a ratio of count of database block addresses to total number of leaf blocks as a measure of

3 invalid guess-database block addresses.

1   19. The system according to claim 16, wherein the database index is a primary B+tree

2 structure, wherein the system further comprises:

3   a mapping table used to support bitmap indexes.

1   20. The system according to claim 19, further comprising:

2   a bitmap index supported by the mapping table.

1   21. The system according to claim 16, wherein the database index is a primary B+tree

2   structure, wherein the system further comprises:

3   a secondary index structure comprising hybrid row identifiers.

1   22. A computer program product for performing a process for maintaining a database

2   management system, comprising:

3   a computer readable medium; and

4   computer program instructions, recorded on the computer readable medium, executable

5   by a processor, for performing the steps of:

6   during splitting of a leaf block of a primary B+tree recording an address of a newly

7   created leaf block; and

8   maintaining the new address in a list as part of primary B+tree metadata.

1   23. A system for performing a process for maintaining a database management system,

2   comprising:

3   a processor operable to execute computer program instructions; and

4   a memory operable to store computer program instructions executable by the processor,

5   for performing the steps of:

6   during splitting of a leaf block of a primary B+tree recording an address of a newly

7   created leaf block; and

8   maintaining the new address in a list as part of primary B+tree metadata.